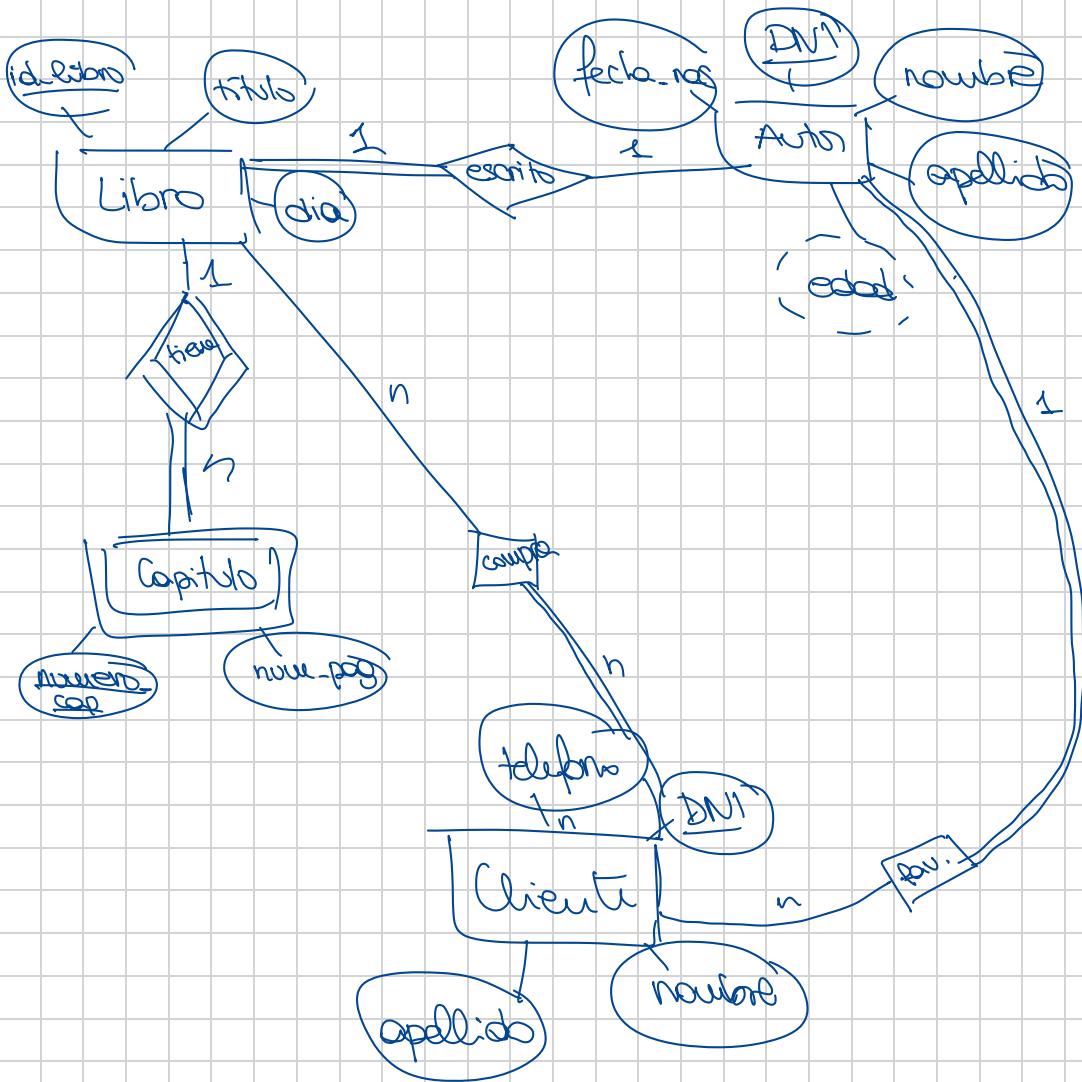


PARTE 1

En la librería de mi barrio han propuesto que cada joven inscrito en la librería seleccione un solo libro para una exposición que se hará el mes que viene. De cada autor sabemos su DNI, nombre, apellido, fecha de nacimiento y edad. Cada libro, el cual tiene un id, título y día de selección, consta de capítulos del que sabemos el número de capítulos que es y el número de páginas que tiene.

En la exposición los clientes de la librería podrán comprar los libros que quieran y como se harán varias copias de cada libro, cada libro puede ser comprado por varios clientes.

Al final cada cliente tendrá que decir cuál es su autor favorito



libro(id-libro, titulo, dia , DNI-autor*)

Autor(DNI-autor , nombre-autor, apellido-autor,fecha-nac.)

Clientes(DNI-cliente , nombre-cliente, apellido-cliente,DNI-autor*)

Capitulo(numero-cap , num-pag, id-libro*)

Teléfono(id-telefono, teléfono, DNI-cliente *)

libro-cliente(id-lib-cliente , id-telefono*, DNI-cliente *)

```
CREATE TABLE IF NOT EXISTS Autor (
    DNI-autor VARCHAR (10) PRIMARY KEY,
    nombre-autor VARCHAR (50),
    apellido-autor VARCHAR (50),
    fecha-nac TIMESTAMP
);
```

```
CREATE TABLE IF NOT EXISTS Libro (
    id-libro INT PRIMARY KEY,
    titulo VARCHAR (100),
    dia INT,
    DNI-autor VARCHAR (10),
    FOREIGN KEY (DNI-autor) REFERENCES Autor (DNI-autor)
);
```

```
CREATE TABLE IF NOT EXISTS Cliente (
    DNI-cliente VARCHAR (10) PRIMARY KEY,
    nombre-cliente VARCHAR (50),
    apellido-cliente VARCHAR (50),
    DNI-autor VARCHAR (10),
    FOREIGN KEY (DNI-autor) REFERENCES Autor (DNI-autor)
);
```

```
CREATE TABLE IF NOT EXISTS Capitulo (
    numero-cap INT,
    id-libro INT,
    num-pag INT,
    PRIMARY KEY (numero-cap, id-libro),
    FOREIGN KEY (id-libro) REFERENCES Libro (id-libro)
);
```

```
CREATE TABLE IF NOT EXISTS Telefono (
    id_telefono INT PRIMARY KEY,
    telefono INT,
    DNI_cliente VARCHAR (10),
    FOREIGN KEY (DNI_cliente) REFERENCES Cliente (DNI_cliente)
);
```

```
CREATE TABLE IF NOT EXISTS Libro_Cliente (
    id_lib_cliente INT PRIMARY KEY,
    id_libro INT,
    DNI_cliente VARCHAR (10),
    FOREIGN KEY (id_libro) REFERENCES Libro (id_libro)
    FOREIGN KEY (DNI_cliente) REFERENCES Cliente (DNI_cliente)
);
```

Insertar datos en la tabla Autor

```
INSERT INTO Autor (DNI_autor, nombre_autor,  
apellido_autor, fecha_nac) VALUES ('12345678A',  
'Gabriel', 'García Márquez', '1927-03-06');  
INSERT INTO Autor (DNI_autor, nombre_autor,  
apellido_autor, fecha_nac) VALUES ('87654321B', 'J.K.',  
'Rowling', '1965-07-31');  
INSERT INTO Autor (DNI_autor, nombre_autor,  
apellido_autor, fecha_nac) VALUES ('56781234C',  
'George', 'Orwell', '1903-06-25');  
INSERT INTO Autor (DNI_autor, nombre_autor,  
apellido_autor, fecha_nac) VALUES ('98765432D',  
'Miguel', 'de Cervantes', '1547-09-29');  
INSERT INTO Autor (DNI_autor, nombre_autor,  
apellido_autor, fecha_nac) VALUES ('23456789E', 'Julio',  
'Cortázar', '1914-08-26');  
INSERT INTO Autor (DNI_autor, nombre_autor,  
apellido_autor, fecha_nac) VALUES ('34567890F', 'Jane',  
'Austen', '1775-12-16');  
INSERT INTO Autor (DNI_autor, nombre_autor,  
apellido_autor, fecha_nac) VALUES ('45678901G',  
'Fiodor', 'Dostoievski', '1821-11-11');  
INSERT INTO Autor (DNI_autor, nombre_autor,  
apellido_autor, fecha_nac) VALUES ('56789012H',  
'Mark', 'Twain', '1835-11-30');  
INSERT INTO Autor (DNI_autor, nombre_autor,  
apellido_autor, fecha_nac) VALUES ('67890123I', 'Franz',  
'Kafka', '1883-07-03');  
INSERT INTO Autor (DNI_autor, nombre_autor,
```

apellido_autor, fecha_nac) VALUES ('78901234J',
'Virginia', 'Woolf', '1882-01-25');

INSERT INTO Autor (DNI_autor, nombre_autor,
apellido_autor, fecha_nac) VALUES ('89012345K',
'Herman', 'Melville', '1819-08-01');

INSERT INTO Autor (DNI_autor, nombre_autor,
apellido_autor, fecha_nac) VALUES ('90123456L', 'Mary',
'Shelley', '1797-08-30');

INSERT INTO Autor (DNI_autor, nombre_autor,
apellido_autor, fecha_nac) VALUES ('11223344M',
'Ernest', 'Hemingway', '1899-07-21');

INSERT INTO Autor (DNI_autor, nombre_autor,
apellido_autor, fecha_nac) VALUES ('22334455N', 'H.P.',
'Lovecraft', '1890-08-20');

INSERT INTO Autor (DNI_autor, nombre_autor,
apellido_autor, fecha_nac) VALUES ('33445566O',
'Stephen', 'King', '1947-09-21');

INSERT INTO Autor (DNI_autor, nombre_autor,
apellido_autor, fecha_nac) VALUES ('44556677P',
'Isabel', 'Allende', '1942-08-02');

INSERT INTO Autor (DNI_autor, nombre_autor,
apellido_autor, fecha_nac) VALUES ('55667788Q',
'Haruki', 'Murakami', '1949-01-12');

INSERT INTO Autor (DNI_autor, nombre_autor,
apellido_autor, fecha_nac) VALUES ('66778899R',
'Umberto', 'Eco', '1932-01-05');

INSERT INTO Autor (DNI_autor, nombre_autor,
apellido_autor, fecha_nac) VALUES ('77889900S',
'J.R.R.', 'Tolkien', '1892-01-03');

INSERT INTO Autor (DNI_autor, nombre_autor,
apellido_autor, fecha_nac) VALUES ('88990011T',
'Agatha', 'Christie', '1890-09-15');

Insertar datos en la tabla Libro

```
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (1, 'Cien Años de Soledad', 19, '12345678A');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (2, 'Harry Potter y la Piedra Filosofal', 23,
'87654321B');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (3, '1984', 3, '56781234C');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (4, 'Don Quijote de la Mancha', 14,
'98765432D');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (5, 'Rayuela', 5, '23456789E');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (6, 'Orgullo y Prejuicio', 19, '34567890F');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (7, 'Crimen y Castigo', 18, '45678901G');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (8, 'Las Aventuras de Tom Sawyer', 4,
'56789012H');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (9, 'La Metamorfosis', 27, '67890123I');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (10, 'Al Faro', 13, '78901234J');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (11, 'Moby Dick', 2, '89012345K');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (12, 'Frankenstein', 30, '90123456L');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (13, 'El Viejo y el Mar', 12, '11223344M');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (14, 'Los Mitos de Cthulhu', 12, '22334455N');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
```

```
VALUES (15, 'It', 10, '33445566O');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (16, 'La Casa de los Espíritus', 8, '44556677P');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (17, 'Tokio Blues', 15, '55667788Q');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (18, 'El Nombre de la Rosa', 21, '66778899R');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (19, 'El Señor de los Anillos', 25, '77889900S');
INSERT INTO Libro (id_libro, titulo, dia, DNI_autor)
VALUES (20, 'Asesinato en el Orient Express', 13,
'88990011T');
```

Insertar datos en la tabla Cliente

```
INSERT INTO Cliente (DNI_cliente, nombre_cliente,
apellido_cliente, DNI_autor) VALUES ('11112222A',
'Carlos', 'Pérez', '12345678A');
INSERT INTO Cliente (DNI_cliente, nombre_cliente,
apellido_cliente, DNI_autor) VALUES ('22223333B',
'Laura', 'Gómez', '87654321B');
INSERT INTO Cliente (DNI_cliente, nombre_cliente,
apellido_cliente, DNI_autor) VALUES ('33334444C',
'Pedro', 'Ramírez', '56781234C');
INSERT INTO Cliente (DNI_cliente, nombre_cliente,
apellido_cliente, DNI_autor) VALUES ('44445555D',
'Ana', 'Fernández', '98765432D');
INSERT INTO Cliente (DNI_cliente, nombre_cliente,
apellido_cliente, DNI_autor) VALUES ('55556666E',
'Luis', 'Martínez', '23456789E');
INSERT INTO Cliente (DNI_cliente, nombre_cliente,
apellido_cliente, DNI_autor) VALUES ('66667777F',
'Elena', 'Rodríguez', '34567890F');
INSERT INTO Cliente (DNI_cliente, nombre_cliente,
```

apellido_cliente, DNI_autor) VALUES ('77778888G', 'Javier', 'López', '45678901G');

INSERT INTO Cliente (DNI_cliente, nombre_cliente, apellido_cliente, DNI_autor) VALUES ('88889999H', 'Andrea', 'González', '56789012H');

INSERT INTO Cliente (DNI_cliente, nombre_cliente, apellido_cliente, DNI_autor) VALUES ('99990000I', 'Miguel', 'Hernández', '67890123I');

INSERT INTO Cliente (DNI_cliente, nombre_cliente, apellido_cliente, DNI_autor) VALUES ('10101010J', 'Sofia', 'Castro', '78901234J');

INSERT INTO Cliente (DNI_cliente, nombre_cliente, apellido_cliente, DNI_autor) VALUES ('11111111K', 'Francisco', 'Ortiz', '89012345K');

INSERT INTO Cliente (DNI_cliente, nombre_cliente, apellido_cliente, DNI_autor) VALUES ('12121212L', 'Clara', 'Mendoza', '90123456L');

INSERT INTO Cliente (DNI_cliente, nombre_cliente, apellido_cliente, DNI_autor) VALUES ('13131313M', 'Roberto', 'Navarro', '11223344M');

INSERT INTO Cliente (DNI_cliente, nombre_cliente, apellido_cliente, DNI_autor) VALUES ('14141414N', 'Natalia', 'Silva', '22334455N');

INSERT INTO Cliente (DNI_cliente, nombre_cliente, apellido_cliente, DNI_autor) VALUES ('15151515O', 'Alberto', 'Rivas', '33445566O');

INSERT INTO Cliente (DNI_cliente, nombre_cliente, apellido_cliente, DNI_autor) VALUES ('16161616P', 'Marta', 'Domínguez', '44556677P');

INSERT INTO Cliente (DNI_cliente, nombre_cliente, apellido_cliente, DNI_autor) VALUES ('17171717Q', 'Raúl', 'Vega', '55667788Q');

INSERT INTO Cliente (DNI_cliente, nombre_cliente,

apellido_cliente, DNI_autor) VALUES ('18181818R',
'Daniela', 'Sánchez', '66778899R');

INSERT INTO Cliente (DNI_cliente, nombre_cliente,
apellido_cliente, DNI_autor) VALUES ('19191919S',
'Cristian', 'Figueroa', '77889900S');

Insertar datos en la tabla Capitulo

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (1, 30, 1);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (2, 40, 1);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (3, 35, 1);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (4, 45, 1);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (1, 25, 2);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (2, 35, 2);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (3, 30, 2);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (4, 40, 2);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (1, 50, 3);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (2, 45, 3);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (3, 40, 3);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (4, 30, 3);

INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (1, 32, 4);

```
INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (2, 38, 4);
INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (3, 44, 4);
INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (4, 50, 4);
INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (1, 28, 5);
INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (2, 36, 5);
INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (3, 42, 5);
INSERT INTO Capitulo (numero_cap, num_pag, id_libro)
VALUES (4, 48, 5);
```

Insertar datos en la tabla Telefono

```
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (1, '123456789', '11112222A');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (2, '987654321', '22223333B');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (3, '555666777', '33334444C');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (4, '777888999', '44445555D');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (5, '222333444', '55556666E');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (6, '888999000', '66667777F');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (7, '666777888', '77778888G');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (8, '111222333', '88889999H');
INSERT INTO Telefono (id_telefono, telefono,
```

DNI_cliente) VALUES (9, '444555666', '99990000I');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (10, '999000111', '10101010J');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (11, '222444666', '11111111K');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (12, '333555777', '12121212L');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (13, '555777999', '13131313M');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (14, '777999111', '14141414N');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (15, '888111222', '15151515O');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (16, '999222333', '16161616P');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (17, '111333555', '17171717Q');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (18, '222555777', '18181818R');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (19, '333777999', '19191919S');

INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (1, '123456789', '11112222A');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (2, '987654321', '22223333B');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (3, '555666777', '33334444C');
INSERT INTO Telefono (id_telefono, telefono,
DNI_cliente) VALUES (4, '777888999', '44445555D');

```
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (5, '222333444', '55556666E');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (6, '888999000', '66667777F');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (7, '666777888', '77778888G');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (8, '111222333', '88889999H');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (9, '444555666', '99990000I');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (10, '999000111', '10101010J');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (11, '222444666', '11111111K');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (12, '333555777', '12121212L');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (13, '555777999', '13131313M');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (14, '777999111', '14141414N');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (15, '888111222', '15151515O');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (16, '999222333', '16161616P');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (17, '111333555', '17171717Q');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (18, '222555777', '18181818R');  
INSERT INTO Telefono (id_telefono, telefono,  
DNI_cliente) VALUES (19, '333777999', '19191919S');
```

Insertar datos en la tabla Libro_Cliente

```
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
```

DNI_cliente) VALUES (1, 1, '11112222A');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (2, 2, '22223333B');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (3, 3, '33334444C');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (4, 4, '44445555D');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (5, 5, '55556666E');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (6, 6, '66667777F');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (7, 7, '77778888G');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (8, 8, '88889999H');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (9, 9, '99990000I');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (10, 10, '10101010J');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (11, 11, '11111111K');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (12, 12, '12121212L');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (13, 13, '13131313M');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (14, 14, '14141414N');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (15, 15, '15151515O');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (16, 16, '16161616P');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (17, 17, '17171717Q');

```
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (18, 18, '18181818R');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (19, 19, '19191919S');
INSERT INTO Libro_Cliente (id_lib_cliente, id_libro,
DNI_cliente) VALUES (20, 20, '22223333B'); -- Cliente
comprando otro libro
```

PROYECTO

1

1. Cliente

ver

El cliente tiene acceso a sus datos y puede ver la información del libro incluido la información de los capítulos, excepto el día de selección del libro. Por otro lado, del autor solo puede saber su nombre y su apellido.

libro(id-libro, título)

Autor(nombre-autor, apellido-autor)

Cliente(DNI-cliente, nombre-cliente, apellido-cliente)

Capítulo(número-cap, num-pag, id-libro*)

Teléfono(id-teléfono, teléfono, DNI-cliente*)

libro-cliente(id-lib-diente, id-libro*, DNI-cliente*)

library employee

Libro (id-libro, título, dia, DNI-autor*)

Autor (DNI-autor [nombre-autor, apellido-autor, fecha-nac])

Cliente (DNI-cliente, nombre-cliente, apellido-cliente, DNI-autor*)

Capítulo (numero-cap , num-pag, id-libro*)

Teléfono (id-teléfono, teléfono, DNI-cliente*)

Libro-Cliente (id-lib-cliente, id-libro*, DNI-cliente*)

El trabajador tiene acceso a ver toda la base de datos y
puede hacer modificaciones excepto en el nombre, apellido
y fecha de nacimiento del autor.

La vista externa se queda igual porque puede ver todos
excepto los atributos nombre-autor, apellido-autor y fecha-nac
no podrán ser modificados.

TRABAJO TECNOLOGIA DE LA INFORMACIÓN

1.2: crear usuarios: captura pantalla del comando SQL para crear cada usuario. (CLIENTE)

The screenshot shows a window titled "Group Role - Login/Group Roles". The tab "SQL" is selected. The code area contains the following SQL command:

```
1 CREATE ROLE cliente WITH
2     LOGIN
3     NOSUPERUSER
4     NOCREATEDB
5     NOCREATEROLE
6     INHERIT
7     NOREPLICATION
8     CONNECTION LIMIT -1
9     PASSWORD 'xxxxxx';
```

At the bottom left are two buttons: an info icon and a question mark icon. At the bottom right are three buttons: "Close" (with an X), "Reset" (with a circular arrow), and "Save" (with a disk icon).

ejemplo creando una vista

The screenshot shows a PostgreSQL 'Create - View' dialog box. The title bar says 'Create - View'. Below it is a tab bar with 'General', 'Definition', 'Code', 'Security', and 'SQL', where 'SQL' is selected. The main area contains the following SQL code:

```
1 CREATE VIEW public.view_cliente_libro
2 AS
3 SELECT id_libro, titulo FROM Libro;
4
5 ALTER TABLE public.view_cliente_libro
6     OWNER TO postgres;
7
8 GRANT SELECT ON TABLE public.view_cliente_libro TO cli
```

At the bottom of the dialog are several buttons: 'Close' (with an X icon), 'Reset' (with a circular arrow icon), and 'Save' (with a disk icon).

libreria/cliente@servidor_cliente

No limit

Query History

Scratch Pad

1 `SELECT * FROM view_cliente_libro;`

Data Output

	<code>id_libro</code>	<code>titulo</code>
1	1	Cien Años de
2	2	Harry Potter y la
3	3	1984
4	4	Don Quijote de la
5	5	Rayuela

Total rows: 20 of 20 Query complete 00:00:00.562

ejemplo dando permiso a una columna en concreto

The screenshot shows a database management interface with a central SQL editor window. The window title is "nombre_autor". At the top, there are tabs: General, Definition, Constraints, Variables, Security, and SQL. The SQL tab is selected, indicated by a blue underline. Below the tabs, a single line of SQL code is displayed:

```
1 GRANT SELECT(nombre_autor) ON public.autor TO cliente;
```

At the bottom of the editor window, there are several buttons: an information icon (i), a question mark icon (?), a close button (X Close), a reset button (Reset), and a save button (Save).

libreria/cliente@servidor_cliente

No limit

Query History

```
1 SELECT nombre_autor FROM Autor;
```

Data Output Messages Notifications

	nombre_autor
1	Gabriel
2	J.K.
3	George
4	Miguel
5	Julio

Total rows: 20 of 20 Query complete 00:00:00 484

apellido_autor

X

General Definition Constraints Variables Security SQL

Privileges

+

Grantee	Privileges	Grantor
cliente	r	postgres

Security labels

+

Provider	Security label



Close

Reset

Save

Query Query History

```
1 SELECT apellido_autor FROM Autor;
```

Data Output Messages Notifications



	apellido_autor character varying (50)	🔒
1	García Márquez	
2	Rowling	
3	Orwell	
4	de Cervantes	
5	Cortázar	
...	...	

Total rows: 20 of 20 | Query complete 00:00:00 604

ejemplo de error a lo que no tiene permiso

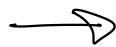
The screenshot shows a MySQL Workbench interface. The connection is set to 'libreria/cliente@servidor_cliente'. The toolbar includes various icons for database management. Below the toolbar, there are tabs for 'Query' and 'Query History', with 'Query' being the active tab. A single query is present in the editor:

```
1 SELECT dni_autor FROM Autor;
```

Below the editor, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is selected and displays the following error message:

```
ERROR: permission denied for table autor  
SQL state: 42501
```

ejemplo de dar permiso a una tabla entera



capítulo

General Columns Advanced Constraints Parameters Security SQL

Privileges

Grantee	Privileges	Grantor
cliente	r	postgres

Security labels

Provider	Security label



Close

Reset

Save

Query Query History

1 `SELECT * FROM Capitulo;`

Execute script
F5

Data Output Messages Notifications



	numero_cap [PK] integer	id_libro [PK] integer	num_pag integer
1	1	1	30
2	2	1	40
3	3	1	35
4	4	1	45
5	1	2	25
6	2	2	25

Total rows: 20 of 20 Query complete 00:00:00.336

Se ha hecho lo mismo con el usuario trabajador

El trabajador puede modificar y ver la tabla libro



libro

General Columns Advanced Constraints Parameters Security SQL

Privileges

Grantee	Privileges	Grantor
trabajador	arw <input type="checkbox"/> ALL <input type="checkbox"/> WITH GRANT OPTION <input checked="" type="checkbox"/> INSERT <input type="checkbox"/> WITH GRANT OPTION <input checked="" type="checkbox"/> SELECT <input type="checkbox"/> WITH GRANT OPTION <input checked="" type="checkbox"/> UPDATE <input type="checkbox"/> WITH GRANT OPTION <input type="checkbox"/> DELETE <input type="checkbox"/> WITH GRANT OPTION <input type="checkbox"/> TRUNCATE <input type="checkbox"/> WITH GRANT OPTION <input type="checkbox"/> REFERENCES <input type="checkbox"/> WITH GRANT OPTION <input type="checkbox"/> TRIGGER <input type="checkbox"/> WITH GRANT OPTION	postgres

Security labels

Provider	Security label

Close Reset Save

Query Query History

```
1 SELECT * FROM libro;
```

Execute script

F5

Data Output Messages Notifications

	id_libro [PK] integer	titulo character varying (100)	dia integer	dni_autor character varying (10)
1	1	Cien Años de	19	12345678A
2	2	Harry Potter y la	23	87654321B
3	3	1984	3	56781234C
4	4	Don Quijote de la	14	98765432D
5	5	Rayuela	5	23456789E
		Oncella o Destituta	10	04567800F

Total rows: 20 of 20 Query completed 00:00:00.257

así con el resto de tablas menos con la de autor ya que no puede modificar el nombre, el apellido y la fecha de nacimiento, se realiza por columnas.

atributo nuevo calculado a partir de atributo compuesto

view_edad

X

General Definition Code Security SQL

```
1  SELECT nombre_autor,
2      apellido_autor,
3      age('2025-01-01 00:00:00'::timestamp without time zone, fecha_nac) AS edad
4  FROM autor;
```

X Close  Save

Query Query history

1 **SELECT * FROM view_edad;**

Data Output Messages Notifications



	nombre_autor character varying (50)	apellido_autor character varying (50)	edad interval
1	Gabriel	García Márquez	97 years 9 mons 26 days
2	J.K.	Rowling	59 years 5 mons 1 day
3	George	Orwell	121 years 6 mons 6 days
4	Miguel	de Cervantes	477 years 3 mons 2 days
5	Julio	Cortázar	110 years 4 mons 6 days
6	Luis	Auster	240 years 16 days

Total rows: 20 of 20 Query complete 00:00:00.302

Library Employee 1	Client	Library Employee 2
<pre> BEGIN; SELECT * FROM Libro; BEGIN; UPDATE Libro SET titolo='Il nome della rosa' WHERE id_libro=18; SELECT * FROM Cliente WHERE DNI_Cliente='11112222A'; SELECT * FROM Libro; UPDATE Cliente SET nombre_cliente='Javier Carlos' WHERE DNI_Cliente='11112222A'; END; </pre>	<pre> BEGIN; SELECT * FROM Libro; SELECT * FROM Cliente WHERE DNI_Cliente='11112222A'; </pre>	<pre> BEGIN; SELECT * FROM Cliente WHERE DNI_Cliente='11112222A'; UPDATE Cliente SET nombre_cliente='Carlos Javier' AND DNI_Cliente='11113333A' WHERE DNI_Cliente='11112222A'; END; UPDATE Cliente SET nombre_cliente='Javier' WHERE DNI_Cliente='11112222A'; END; </pre>

2. False. Un schedule es serial cuando las transacciones se ejecutan sin intercalarse, en este caso las transacciones se ejecutan intercaladamente y esa es la única razón por la que no es serial. Por lo tanto, el que sea serializable no es la razón por la que el schedule no es serial.
2. lost update Problem: el trabajador 1 selecciona un cliente, y el trabajador 2 selecciona el mismo cliente, después el trabajador 1 actualiza el nombre del cliente, el trabajador 2 realiza también la actualización del nombre del mismo cliente pero sobre la selección que ha hecho antes de la actualización realizada por el trabajador 1, por lo que se pierde la actualización hecha por el trabajador 1 porque la última actualización del trabajador 2 sobrescribe la anterior del trabajador 1.

- **Unrepeatable Read Problem:** el cliente hace un select de la tabla libros y obtiene una respuesta, el trabajador 1 actualiza la información de un libro, y el cliente al volver a hacer el mismo select obtiene otra respuesta diferente, esto es un problema porque se espera que los datos permanezcan constantes durante toda la transacción.
- **Phantom Read Problem:** el trabajador 2 actualiza el DNI del cliente y después el cliente intenta actualizar el nombre correspondiente al DNI anterior el cual ya no existe porque ha sido modificado; por lo tanto se produce un error.

3. Library Employee 1

Client

Library Employee 2

BEGIN

UPDATE Libro SET titulo = 'El
name della rosa' WHERE id_Libro = 18;

SELECT * FROM Clientes WHERE
DNI_Cliente = '111112222A';

UPDATE Cliente SET nombre_ -
cliente = 'Javier Gordo' WHERE
DNI_Cliente = '111112222A'

END;

BEGIN

SELECT * FROM Libros;

SELECT * FROM Libros;

UPDATE Cliente SET nombre_ -
cliente = 'Javier' WHERE
DNI_Cliente = '111112222A'

END;

SELECT * FROM Clientes WHERE
DNI_Cliente = '111112222A'

FOR UPDATE;

• UPDATE Cliente SET nombre_ -
cliente = 'Gordo Javier' AND
DNI_Cliente = '44413333A' WHERE
DNI_Cliente = '111112222A'

END;

4 Es serial porque las transacciones se ejecutan una tras otra sin
intercambio.

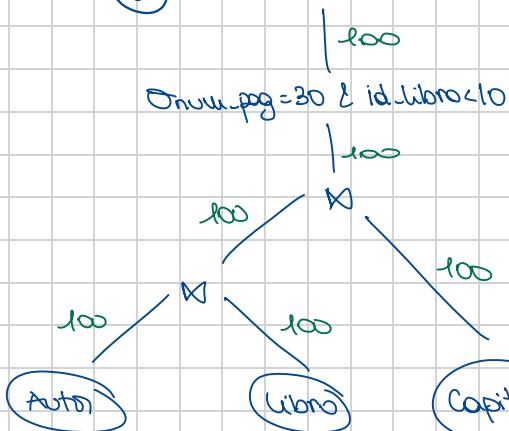
3) `SELECT DISTINCT título, nombre_autor WHERE num_pag = 30 AND id_libro < 10 FROM Autor NATURAL JOIN Libro NATURAL JOIN Capítulo`

1. El select y la variable nombre-autor están mal escritos y el orden de la consulta no es correcto, primero va el FROM y luego el WHERE.
2. Error Execution : falta de ortografía en el Select, está escrito Select y tiene que ser select y variable mal escrita, se llama nombre-autor no nombre-auto.
Error Parser : el orden de la consulta no es correcto, FROM va antes que WHERE.
3. `SELECT DISTINCT título, nombre-autor FROM Autor NATURAL JOIN Libro NATURAL JOIN Capítulo WHERE num_pag = 30 AND id_libro < 10.`

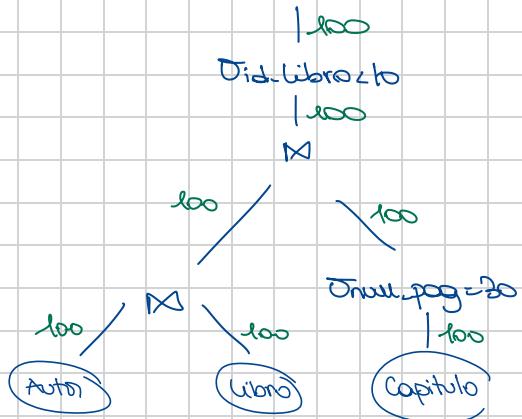
3. SELECT DISTINCT titulo, nombre_autor FROM Autor NATURAL JOIN Libro NATURAL JOIN Capitulo WHERE num_pag = 30 AND id_libro < 10.

4.

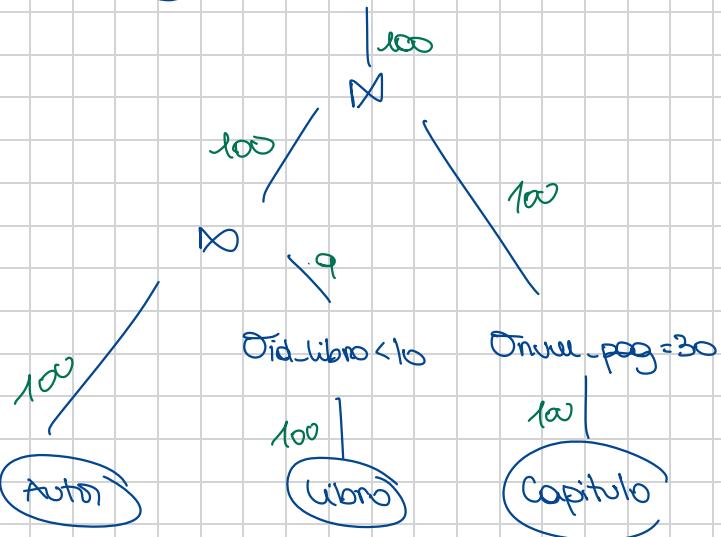
(1) T(titulo, nombre_autor)



(2) T(titulo, nombre_autor)



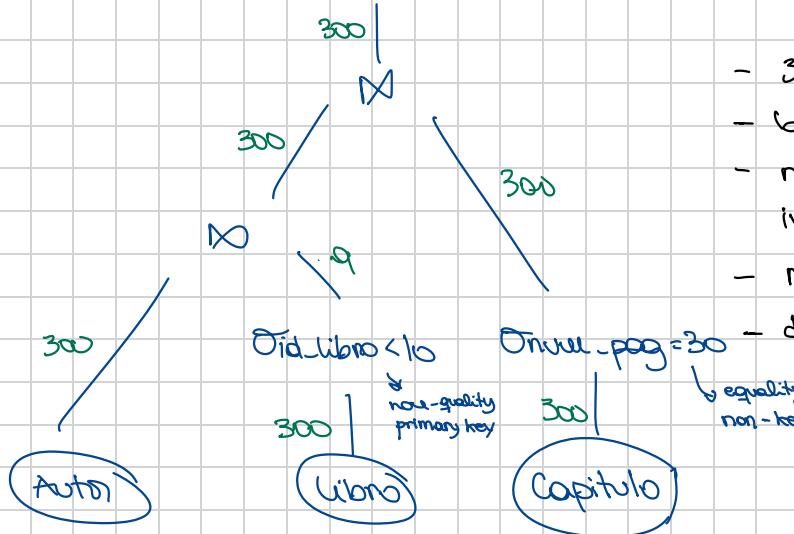
(3) T(titulo, nombre_autor)



El más optimizado es el 3 ya que primero hace las selecciones de las tablas, siendo más probable que se reduzca el número de filas, y después hace el natural join con tablas más reducidas por lo que tarda menos tiempo computacionalmente.

5.

Title, nombre, autor



$$n_{libros} = 300$$

$$brf_{libros} = 3$$

$$b_{libros} = \frac{300}{3} = 100$$

$$x = 3$$

$$n_{libros} = 300$$

$$SL = l/d = l/300$$

$$S = SL * r = \frac{1}{300} \cdot 300 = 1$$

$$n_{capitulos} = 300$$

$$brf_{capitulos} = 3$$

$$b_{capitulos} = \frac{300}{3} = 100$$

$$x = 3$$

$$n_{libros} = 300$$

$$SL = l/d = l/300$$

$$S = SL * r = \frac{1}{300} \cdot 300 = 1$$

- 300 records
- blocking factor 3 = bfr
- number of level of an index 3 = x
- number of buffers 3 = M

- data ordered

- equality
non-key



Full table scan or Brute Force : $b = 300$ ms

non-quality condition on a primary key

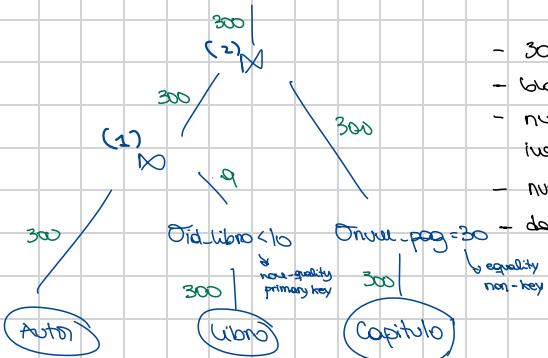
$$x + (b/2) = 3 + \frac{100}{2} = 53 \text{ ms}$$

Full table scan or Brute Force : $b = 100$ ms

Equality condition on non-key : $x + (b/2) =$

$$= 3 + \frac{100}{2} = 53 \text{ ms}$$

Titulo, nombre, autor



- 300 records
- blocking factor 3 = bfr
- number of level of an index 3 = x
- number of buffers 3 = M
- data ordered

NATURAL JOIN (1)

$$rows_r = 300$$

$$buf_author = 3$$

$$buf_autor = \frac{300}{3} = 100$$

$$M = 3$$

$$r_{selection_id} = 9$$

$$bfr_{selection_id} = 3$$

$$bfr_{selection_id} = 3$$

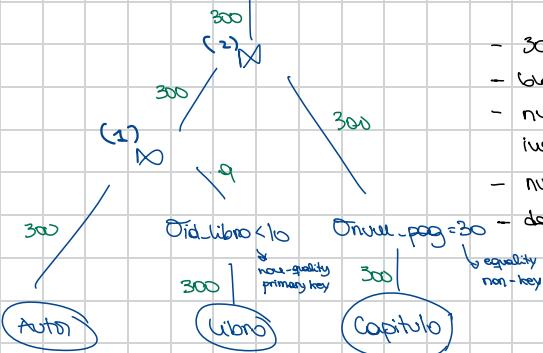
$$\text{Nested Loop (Records)}: b^R + (r^R * b^S) = 100 + (300 * 3) = 3000 \text{ ms}$$

$$\text{Nested Loop (Blocks)}: b^R + (b^R / (M-2)) * b^S = 100 + (100 / (3-2)) * 3 = \\ = 400 \text{ ms}$$

$$\text{Sort Merge: } b^R + b^S + \text{Cost to sort} = 100 + 3 + 0 = 103 \text{ ms}$$

(data is ordered)

Títulos, autores, auto



- 300 records
- blocking factor 3 = ldr
- number of level of an index 3 = X
- number of buffers 3 = M
- data ordered

NATURAL JOIN (2)

$$\begin{cases}
 \text{root_join1} = 300 \\
 \text{bf_root_join1} = 3 \\
 \text{bf_root_join1} = \frac{300}{3} = 100
 \end{cases}
 \quad
 \begin{cases}
 \text{selection_num1} = 300 \\
 \text{bf_selection_num1} = 3 \\
 \text{bf_selection_num1} = \frac{300}{3} = 100
 \end{cases}
 \quad
 \left\{ \begin{array}{l} \text{S} \\ \text{J} \end{array} \right.$$

$$M = 3$$

$$\text{Nested Loop (Records)}: b^R + (r^R * b^S) = 100 + (300 * 100) = 30100 \text{ ms}$$

$$\begin{aligned}
 \text{Nested Loop (Blocks)}: b^R + (b^R / (M-2)) * b^S &= 100 + (100 / (3-2)) * 100 = \\
 &= 10100 \text{ ms}
 \end{aligned}$$

$$\begin{aligned}
 \text{Sort Merge: } b^R + b^S + \text{Cost to sort} &= 100 + 100 + 0 = 200 \text{ ms} \\
 &\quad (\text{data is ordered})
 \end{aligned}$$

$$\text{Plan ejecucional: } 53 + 53 + 103 + 200 = 409 \text{ ms}$$

6. Tres realizar los diferentes cálculos, lo más óptimo es: hacer un non-equality condition on a primary key (53 ms) en el Tid-libro < 10 ; hacer un equality condition on a non-key (53 ms) en el Onivel.pag = 30 ; en el primer Natural Join un Sort Merge (103 ms) y en el segundo Natural Join un Sort Merge (200 ms) haciendo un costo óptimo de 409 ms.